Rolling Up Random Variables in Data Cubes

Phillip M. Yelland*

Abstract
Data cubes, first developed in the context of on-line analytic processing (OLAP) applications for databases, have become increasingly widespread as a means of structuring data aggregations in other contexts. For example, increasing levels of aggregation in a data cube can be used to impose a hierarchical structure—often referred to as *roll-ups*—on sets of cross-categorized values, producing a summary description that takes advantage of commonalities within the cube categories. In this paper, we describe a novel technique for realizing such a hierarchical structure in a data cube containing discrete random variables. Using a generalization of an approach due to Chow and Liu, this technique construes roll-ups as parsimonious approximations to the joint distribution of the variables in terms of the aggregation structure of the cube. The technique is illustrated using a real-life application that involves monitoring and reporting anomalies in Web traffic streams over time.

Key Words: Data cube, roll up, Chow-Liu.

1. Introduction

A query submitted to a search engine such as Google may classified according to a wide range of associated categorical attributes, many of which may be established by inspection of the query itself, or of the HTTP request initiating the query. Examples of such attributes include the likely country or geographical region where the query originated, the language in which the query is composed and the type of browser used to issue the query. Thus by examining server logs, a cross tabulation may be compiled, totaling queries with particular attributes that arrive during the course of a day. An illustration of entries in such a cross-tabulation is given in table 1. Here, each query total is juxtaposed with the particular combination of attribute values—the *breakdown*—to which it applies.[1]

Country	Language	Browser	Total Queries
Germany	German	Firefox	3,000,000
Germany	German	Chrome	3,000,000
Germany	English	Chrome	600,000
Brazil	Portuguese (Br)	Chrome	80,000
Brazil	English	Chrome	20,000
Brazil	English	Firefox	20,000

Table 1: A fictive sample cross-tabulation of query totals

Given a set of cross tabulated query totals like those in table 1, we might consider further aggregating them. For example, we can calculate total queries from Germany in German issued using *any* browser in table 1, by adding together totals for the breakdowns (*Germany, German, Firefox*) and (*Germany, German, Chrome*). The resulting sum is labeled with the breakdown (*Germany, German,* $\star$), where the "$\star$" symbol (the equivalent of Gray et al's "`ALL`") indicates aggregation over all values of *Browser*. These aggregates, together with the original breakdowns and totals themselves, comprise a *data cube* in Gray et al.'s terminology.

The examples in table 1 represent query totals for a single day; repeating the process

*Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA.

[1]To protect confidentiality, query totals throughout this paper are fictive, and purely for the purpose of illustration.

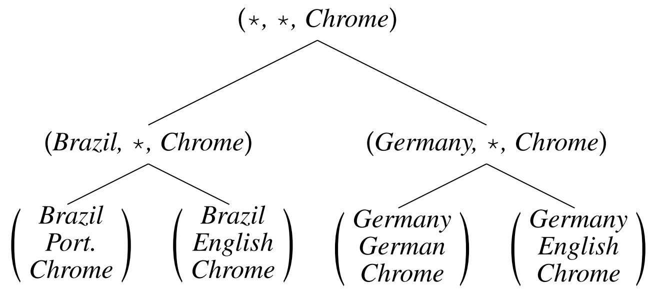

Figure 1: A sample roll-up

over consecutive days produces a time series of totals associated with each breakdown. The inspiration for the work detailed in this paper derives from a collection of some 250,000 such time series describing queries submitted to Google over time. To help detect problems in the site's hardware or software, logging problems or spam attacks, we examine this collection of time series using statistical models to uncover *anomalies*—daily totals that depart radically from expected norms. The anomalies exposed are then passed to analysts for manual inspection, since accurate diagnosis of the underlying causes generally requires extensive knowledge of the site's operational infrastructure. In designing software to support this monitoring process, we found that aside from the actual detection of anomalies, reporting them accessibly constituted a substnatial challenge. With some 250,000 series under examination, even if anomalies are detected in 1% of observations per day, we can still expect thousands of anomalies to be reported in the course of a week—too large a data set to be manually inspected without further processing. It is the framework we developed for organizing and digesting the detected anomalies that is the subject of this paper.

We began by observing that an effective way of organizing anomalies takes advantage of the data cube breakdowns with which they are associated. Using the breakdowns in table 1 as an example, we might note that anomalies were reported in (the time series labeled with) breakdowns (*Germany, German, Chrome*) and (*Germany, English, Chrome*). If we "aggregate" these anomaly reports (a procedure we explore more fully in later sections), we can associate this aggregated report with the more general breakdown (*Germany*, $\star$, *Chrome*). Likewise, we can aggregate anomalies in (*Brazil, Portuguese (Br), Chrome*) and (*Brazil, English, Chrome*), affixing the result to the breakdown (*Brazil*, $\star$, *Chrome*). This process may be repeated, of course, with the more general breakdowns: By aggregating the reports in (*Germany*, $\star$, *Chrome*), we produce a report for the breakdown ($\star$, $\star$, *Chrome*). The upshot of this procedure—which we refer to in its entirety as a *roll-up* of the original anomalies—is illustrated in figure 1, where a line depicts the association between a breakdown and its immediate aggregation in the roll-up.

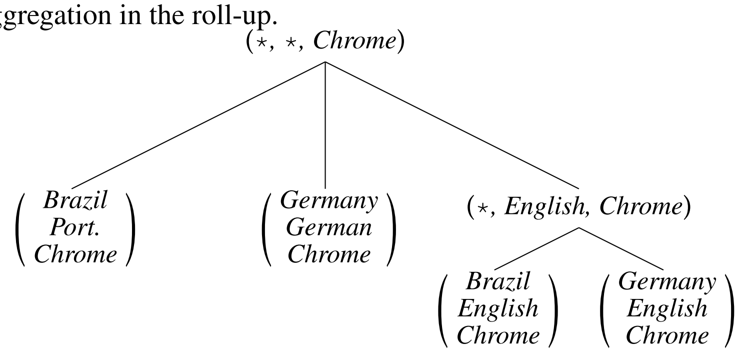

Figure 2: Another roll-up on the same breakdowns as figure 1

Rolling up anomalies in this way can greatly reduce the number of breakdowns that need to be manually examined. In general, the analyst can proceed down a roll-up, from the most general breakdowns to their more numerous component breakdowns; in many cases, any problem manifest by anomalies in the component breakdowns will be evinced in the aggregated reports in the more general ones, obviating the need to examine each of the component breakdowns in turn.

An issue that arises immediately when we begin assembling roll-ups of anomalies is that almost invariably, there are many different roll-ups that can be constructed on the same collection of basic breakdowns. Figure 2, for example, shows a different roll-up based on the same breakdowns as that in figure 1. Ideally, we seek a formal, intuitively-appealing criterion according to which the merits of competing roll-ups can be assessed. Following a brief review of related work in the next section, the remainder of this paper is devoted to devising such a criterion and a procedure for producing roll-ups that satisfy it optimally.

2. Related Work

Data (or OLAP-) cubes rose to prominence during the 1990's, with seminal articles from the period by Gray et al. (1997) and Codd et al. (1993). Data cubes offered to facilitate the use of large data sets in decision support by providing the ability to deal with cross-categorized or "multi-dimensional" data at different levels of aggregation. Fundamental to this ability are the two operations *drill-down*—disaggregating data by introducing a new category/dimension or a finer resolution for an existing one— and *roll-up*—aggregating data by coarsening or eliding dimensions. (The latter term, of course, we have appropriated to label the structures we seek in this paper.)

As the use of data cubes became more widespread, attempts were made to explore their capabilities in a formal setting. Early works in this area include (Agrawal et al. 1997) and (Gyssens and Lakshmanan 1997), both of which propose algebras to characterize operations on data cubes, along the lines of Codd's (1970) earlier formalization of relational databases. Later authors began to draw on order-theoretic concepts—already featured in the literature on the implementation of data cubes (Harinarayan et al. 1996)—in formal descriptions of the technology. Torlone (2003), for example, uses a partial order he terms a *roll-up relation* to model levels of aggregation in a data cube, and Fagin et al. (2005a) incorporate lattices into their definition of *multi-dimensional databases* with a similar purpose.

Fagin et al.'s (2005a) work—like the related (Fagin et al. 2005b)—also typifies a strand of investigation that seeks to characterize operations on a data cube in terms of constrained combinatorial optimization problems, much as we do in later sections of this paper. They describe operations on data cubes which identify sets of (generally aggregate) breakdowns that optimally cluster, summarize or distinguish sets of values in a cube. In a similar vein, Sarawagi (1999) details a related procedure that chooses breakdowns to minimize entropy, yielding an optimally compact representation of one data cube in terms of another—particularly applicable when seeking to characterize changes in a data cube over time.

This quest for optimal compact representations in data cubes is further pursued by Agarwal et al. (2007), whose work closely parallels that described here. They seek a tree of aggregated breakdowns in a data cube that most "parsimoniously" (a term they define formally) account for the changes in values at its leaves. Agarwal et al.'s (2007) work differs from ours in that it: *a*) seeks to describe changes in non-stochastic cube values, not random variables; *b*) combines a heuristic quantification of "parsimony" with an error measure to constitute its optimization criterion, rather than relying on a single measure (Kullback-Leibler divergence) as we do; *c*) assumes a fixed set of breakdowns that enter into the solution, whereas we select breakdowns as part of the solution procedure.

3. Notation

To begin, let Σ be the set of all *symbols* that may appear in breakdowns. We assume that this set includes the element $*$ (the "all" symbol). A flat ordered set $\langle \Sigma, \sqsubseteq \rangle$ results if we take $*$ as the "top" element of Σ, so that for $t, s \in \Sigma, t \sqsubseteq s$ iff $t = s$ or $s = *$. We render $\langle \Sigma, \sqsubseteq \rangle$ a *join-semilattice* (Davey and Priestley 2002, p. 170) by equipping it with a *join* operator, $\sqcup$, which yields the least upper bound of its arguments. For $t, s \in \Sigma,\ t \sqcup s = t$ if $t = s$, and $*$ otherwise. This operation is associative and commutative, so joins extend directly to finite sets of symbols, and we use $\sqcup S$ to denote the join of all symbols in the finite set $S \subseteq \Sigma$.

Breakdowns are simply tuples of symbols with a fixed length, p. The set Δ of all breakdowns is therefore the p-fold product of the symbol set, Σ^p. The partial order $\sqsubseteq$ and join operator $\sqcup$ on symbols are extended to Δ coordinate-wise, making Δ a join-semilattice, like Σ. Thus, for $b, c \in \Delta$:

$$
\begin{aligned}
b \sqsubseteq c \quad & \text{iff} \quad t_i \sqsubseteq s_i, \text{ for } i = 1, \ldots, p, & (1) \\
b \sqcup c \quad & = \quad (t_1 \sqcup s_1, \ldots, t_p \sqcup s_p). & (2)
\end{aligned}
$$

Joins on breakdowns inherit the associativity and commutativity of joins on symbols, and also extend naturally to finite sets; $\sqcup B$ is the join of all breakdowns in the set $B \subseteq \Delta$. In concrete terms, the join of a set of breakdowns is the least general breakdown that aggregates them all. For example:

$$
\bigsqcup \left\{ \begin{array}{c} (\textit{Germany, German, Chrome}), \\ (\textit{Germany, German, Chrome}), \\ (\textit{Brazil, English, Chrome}) \end{array} \right\} = (\star, \star, \textit{Chrome})
$$

In looking for "optimal" roll ups of a set $B \subseteq \Delta$ of breakdowns, we will (at least in principle) be obliged to examine aggregations of every subset of breakdowns in B. This set of aggregations is represented by the set $\mathcal{S}(B)$ of joins of all subsets of B, which in turn (along with the appropriate restrictions of $\sqsubseteq$ and $\sqcup$ from Δ) comprises the *(sub)semilattice generated by* B, the smallest sub-semilattice of Δ containing B (Davey and Priestley 2002, p. 60).

3.1 Graphs of Breakdowns

Since the roll-ups we seek are essentially graphical structures, it is convenient to render in graphical form the lattices of breakdowns—in particular the semilattice $\mathcal{S}(B)$ generated by breakdown set B—in we will look for them:

Definition 1 *The* dual order graph, $\mathcal{G}(B)$ generated by a set of breakdowns B *is the directed graph* (V, A) *with vertices* $V = \mathcal{S}(B)$ *and arc set* A *such that for* $b, c \in \mathcal{S}(B)$, $b \to c \in A$ *iff* $c \sqsubset b$. *Note that the breakdown* $\sqcup B$ *is the root vertex of* $\mathcal{G}(B)$.

Figure 3 illustrates (with obvious abbreviations) the dual order graph for the semilattice generated by the breakdowns featured in figure 1 and 2. As the illustration shows, the nodes in the dual order graph comprise the generating breakdowns and all of their aggregations, while each edge connects a breakdown to those breakdowns (if any) it aggregates. Now a roll up of the given breakdowns of the sort discussed in section 1 can be understood as a tree in the dual order graph that connects each of the original breakdowns—either directly, or through intermediate aggregate breakdowns—to the root node. By way of illustration, figure 3 shows the roll up in figure 2 on the example dual order graph.

Next, we introduce a concept from graph theory literature (Goemans and Myung 1993):

Definition 2 *Given a directed graph* $\mathcal{G} = (V, A)$ *with root vertex* r*, and a set* $T \subseteq V$ *of* terminal vertices*, a* Steiner arborescence *is a tree in* $\mathcal{G}$*, rooted at* r*, that spans* T.

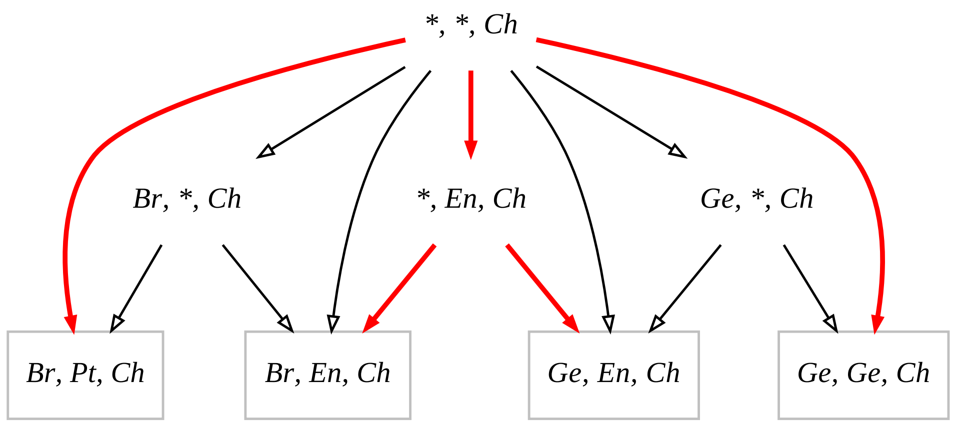

Figure 3: The dual order graph $\mathcal{G}(\{(Br,\ Pt,\ Ch), (Br,\ En,\ Ch), (Ge,\ Ge,\ Ch), (Ge,\ En,\ Ch)\})$, with the roll up in figure 2 (bold, in red, with solid arrow heads).

It is not hard to see that a roll up of a set of breakdowns is a Steiner arborescence on the dual order graph, with terminal vertices determined by the generating breakdowns. Explicitly:

Definition 3 *A* roll-up ρ *on a set of breakdowns B is a Steiner arborescence with terminals B on the dual order graph $\mathcal{G}(B)$ generated by B. We denote by V_ρ the vertices of roll-up ρ (a set of breakdowns which must necessarily include the root breakdown $\sqcup B$). Let Ψ_B be the set of all roll-ups on the breakdown set B.*

We conclude with a definition that will be used later on:

Definition 4 *A roll-up $\rho \in \Psi_B$ defines a function $\mathrm{pa}_\rho : V_\rho \to (V_\rho \cup \{0\})$, where $0 \notin V_\rho$, that maps each vertex in $V_\rho - \{\sqcup B\}$ to its parent, and maps the root breakdown $\sqcup B$ to the distinguished value 0.*

3.2 Anomaly Reports

The Steiner arborescences of the previous section provide a formal description of the tree-structured roll ups we seek, but in order to determine the merit of any particular roll up, we need to associate such Steiner arborescences with the anomaly reports discussed in section 1. To begin, assume that we are monitoring anomalies in a set of breakdowns B, and consider the anomaly report for particular breakdown $b \in B$ on a given day. We characterize anomalies—if any—observed during the day in the timeseries associated with breakdown b as a set, $\mathrm{obs}(b)$, containing encoded values. The precise encoding used may vary from application to application,[2] but for the sake of concreteness, in the scheme used here, values have the following interpretation:[3]

$$\mathrm{obs}(b) = \begin{cases} \emptyset & - \quad \text{No anomalies observed in breakdown } b. \\ \{\mathsf{P}\} & - \quad \text{The timeseries value for } b \text{ is } \textit{above} \text{ what was expected.} \\ \{\mathsf{N}\} & - \quad \text{The timeseries value for } b \text{ is } \textit{below} \text{ what was expected.} \end{cases} \tag{3}$$

For roll ups, as discussed in section 1, we need to prepare *aggregated* anomaly reports. For breakdown $b \in B$, the associated aggregated report is denoted $x\langle b\rangle$, and it describes anomalies observed in all the breakdowns $\{c \in B \mid c \sqsubseteq b\}$ that b aggregates (this latter set is known as b's *down-set*, or *order ideal* (Davey and Priestley 2002)). It is defined simply as the union of all the anomaly reports for breakdowns in the down set:

$$x\langle b\rangle = \bigcup\{\mathrm{obs}(c) \mid c \in B, c \sqsubseteq b\}. \tag{4}$$

[2]So long as a finite set of encoded values is used, the treatment below applies without alteration.

[3]Here, $\emptyset$ denotes the empty set.

Note that $x\langle b\rangle$ may take four distinct (set-valued) values, $\emptyset, \{\mathsf{P}\}, \{\mathsf{N}\}$ or $\{\mathsf{P}, \mathsf{N}\}$, corresponding to observations of no anomalies, all positive, all negative and both positive and negative anomalies in b's down-set, respectively.

Now taking b as fixed, consider observing $x\langle b\rangle$ on different days. For the sake of technical expedience (and in later sections, computational tractability), we assume that occurrences of anomalies within the same breakdown on different are independent. Of course, this is far from true in general, but in practice, we have found that it often suffices as a first approximation, and simplifies matters greatly; we will revisit the topic later in the paper . With this assumption, each day's realization $x\langle b\rangle$ can be viewed as a draw of a random variable, $X\langle b\rangle$, with a discrete probability distribution $P(X\langle b\rangle)$ over the sample space $\{\emptyset, \{\mathsf{P}\}, \{\mathsf{N}\}, \{\mathsf{P}, \mathsf{N}\}\}$.

3.3 Collections of Anomaly Reports

While we assume that anomalies in the same breakdown on different days are independent, we make no such assumptions about anomalies in *different* breakdowns on the *same* day. Indeed, the construction of aggregated anomaly reports in equation (4) means that they are necessarily dependent on the reports for their component breakdowns. And more generally, it is not unreasonable to expect that breakdowns with common breakdown values will manifest similar anomalies at the same time—for example, a problem with the Chrome browser is apt to cause anomalies in component breakdowns in figure 1.

Typically, therefore, we will be concerned with the joint distribution of some vector $(X\langle b_1\rangle, \ldots, X\langle b_n\rangle)$, of random variables associated with a sequence of breakdowns $(b_1, \ldots, b_n)$. While we do not assume that $X\langle b_1\rangle, \ldots, X\langle b_n\rangle$ are independent, the construction of the $X\langle b\rangle$'s as anomaly reports and their aggregations allows us to assume that they are *exchangeable modulo breakdowns*.[4] A special case of McCullagh's (2005) notion of *exchangeability modulo x*, this means that the joint probability of the variables is determined solely by their associated breakdowns, and not by the order in which they appear in the joint vector; formally, for all permutations $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$, $P(X\langle b_1\rangle, \ldots, X\langle b_n\rangle) = P(X\langle b_{\sigma(1)}\rangle, \ldots, X\langle b_{\sigma(n)}\rangle)$. In the interests of brevity, for a set of breakdowns B, we use the expression $X\langle B\rangle$ to denote the set of associated random variables $\{X\langle b\rangle \mid b \in B\}$. The assumption of exchangeability means that we can extend this notation without ambiguity, using $P(X\langle B\rangle)$ to signify the joint probability (distribution) of $X\langle B\rangle$.

Further economy of notation also issues from "implicit marginalization" of joint distributions. Thus, given the distribution $P(X\langle B\rangle)$ and a subset $C \subseteq B$ (and with "\" for set difference), let:

$$P(X\langle C\rangle) = \sum_{X\langle B\setminus C\rangle} P(X\langle B\rangle), \tag{5}$$

where the sum is over all values of the variables in the set $X\langle B\setminus C\rangle$.

4. Optimal Roll Ups

We've seen how anomaly reports may be understood as a collection of random variables, $X\langle B\rangle$, labelled by a set of breakdowns, B. By aggregating anomaly reports, we can extend $X\langle B\rangle$ to the semilattice of breakdowns, $\mathcal{S}(B)$, generated by B, producing an expanded collection of random variables, $X\langle \mathcal{S}(B)\rangle$. Recognizing the partial order between breakdowns in $\mathcal{S}(B)$, we can construct the dual order graph, $\mathcal{G}(B)$, from $\mathcal{S}(B)$; a roll-up is a Steiner

[4] An alternative approach assumes an arbitrary ordering on the set Δ (not necessarily extending $\sqsubseteq$) using the result as an index set for $X\langle\circ\rangle$ *qua* a stochastic process; we then require that the process $X\langle\circ\rangle$ be exchangeable in the usual sense. While conceptually appealing, this approach is technically more involved than using exchangeability modulo breakdowns, as is done here.

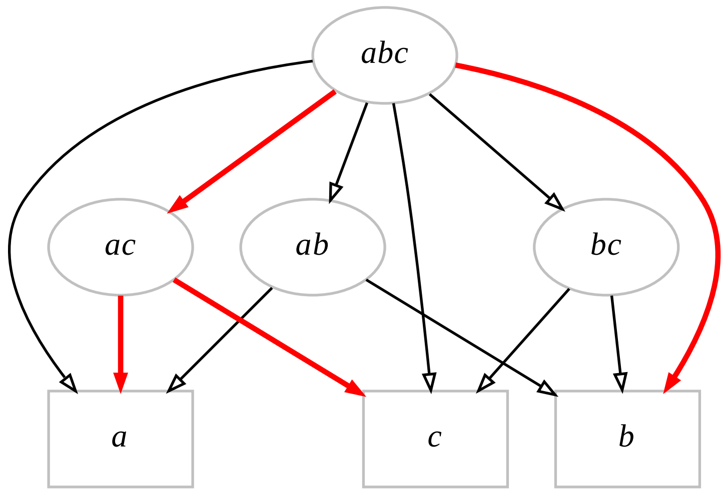

Figure 4: The dual order graph $\mathcal{G}(\{a, b, c\})$.

arborescence on this graph—a rooted tree that encompasses at least the breakdowns in B. Therefore the vertices V_ρ of a roll-up ρ select a set of random variables from the collection $X\langle\mathcal{S}(B)\rangle$. How, then, are we to determine which of the many possible roll-ups on $\mathcal{G}(B)$ is "optimal"?

a	b	c	ab	bc	ac	abc
P	N	∅	PN	N	P	PN
P	P	N	P	PN	PN	PN
P	∅	N	P	P	PN	PN
∅	∅	P	∅	P	P	P
∅	∅	∅	∅	∅	∅	∅
∅	N	∅	P	N	∅	P
N	N	P	N	PN	PN	PN
N	N	∅	N	N	N	N
P	N	∅	PN	N	P	PN

Table 2: Joint sample values of $\mathcal{S}(\{a, b, c\})$

To make the following discussion more concrete, we will introduce a simple example: In figure 4, the dual order graph of three breakdowns a, b and c is depicted,[5] and an example roll-up on this graph is picked out in red. Table 2 displays sample values of the random variables associated with the dual order graph (representing anomalies detected in the corresponding breakdowns on different days) with the observed anomaly reports for breakdowns a, b and c on the left of the table, and the aggregated reports (calculated as described in section 3.3) on the right.[6]

Leaving aside for the moment the aggregated variables, using these sample values, we can estimate the joint distribution of the obeserved random variables simply by counting occurrences of different report values—the result is shown in table 3. The joint distribution can be considered as a description of the observed anomalies (in a real application, we will usually have many more samples than in table 2), relating in summary form the occurence of different types of anomaly in the breakdowns.

Unfortunately, the joint distribution itself will generally grow exponentially with the

[5]For brevity's sake, we have elided the "$\sqcup$" operator in the labels of aggregated breakdowns, writing "ac", for example, instead of "$a \sqcup c$".

[6]Again, we abbreviate $\emptyset, \{\mathsf{P}\}, \{\mathsf{N}\}$ and $\{\mathsf{P}, \mathsf{N}\}$ as $\emptyset, \mathsf{P}, \mathsf{N}$ and PN, resp.

		b		
c	a	N	$\emptyset$	P
N	N	0.00	0.00	0.00
	$\emptyset$	0.00	0.00	0.00
	P	0.00	0.11	0.11
$\emptyset$	N	0.11	0.00	0.00
	$\emptyset$	0.11	0.11	0.00
	P	0.22	0.00	0.00
P	N	0.11	0.00	0.00
	$\emptyset$	0.00	0.11	0.00
	P	0.00	0.00	0.00

Table 3: Joint distribution $P(a, b, c)$

number of observed breakdowns (with just 100 observed breakdowns, a table like (3) would in principle have 5×10^{47} entries), rendering it impractical in real applications. Furthermore, the joint distribution provides scant information about aggregated anomaly reports, and adding them explicitly to the table simply compounds the problem of scalability.

Taking a cue from the work of Chow and Liu (1968) (further developed in (Chow and Wagner 1973)), however, we can view a roll-up such as that highlighted in figure 4 as a means of abbreviating the joint distribution both of the observed breakdowns and a selection of aggregations, too. The roll-up in figure 4, for example, comprises the random variables $X\langle a\rangle, X\langle b\rangle, X\langle c\rangle, X\langle ac\rangle$ and $X\langle abc\rangle$, with joint distribution $P(X\langle a\rangle, X\langle b\rangle, X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle)$. Rather than recording the full joint distribution (which has some 432 entries), however, we can produce an approximation to it by viewing the roll-up as a very simple form of *Bayesian network* (see (Koski and Noble 2009) for further details of the following).

To see how, first note that by the chain rule of probability, we can factorize the full joint distribution of the roll-up's random variables into conditional distributions of its constituents:

$$\begin{aligned} P(X\langle a\rangle, X\langle b\rangle, X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle) = {} & P(X\langle a\rangle | X\langle b\rangle, X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle) \\ & \times P(X\langle b\rangle | X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle) \\ & \times P(X\langle c\rangle | X\langle ac\rangle, X\langle abc\rangle) \\ & \times P(X\langle ac\rangle | X\langle abc\rangle) \\ & \times P(X\langle abc\rangle). \end{aligned}$$

This yields little apparent gain, but we can regard the roll-up as series of *conditional independence* assertions that hold with various degrees of approximation in the data. Informally, the roll-up asserts that the conditional distribution of $X\langle a\rangle$, for example, is (to some approximation) determined solely by its immediate parent in the roll-up, $X\langle ac\rangle$. Formally, $X\langle a\rangle$ is approximately *conditionally independent* of $X\langle b\rangle$, etc., given $X\langle ac\rangle$, so that:

$$\begin{aligned} P(X\langle a\rangle | X\langle b\rangle, X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle) &\approx P(X\langle a\rangle | X\langle ac\rangle), \\ P(X\langle b\rangle | X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle) &\approx P(X\langle b\rangle | X\langle ac\rangle), \\ P(X\langle c\rangle | X\langle ac\rangle, X\langle abc\rangle) &\approx P(X\langle c\rangle | X\langle ac\rangle). \end{aligned}$$

Thus:

$$\begin{aligned} & P(X\langle a\rangle, X\langle b\rangle, X\langle c\rangle, X\langle ac\rangle, X\langle abc\rangle) \approx \\ & \qquad P(X\langle a\rangle | X\langle ac\rangle) P(X\langle b\rangle | X\langle ac\rangle) P(X\langle c\rangle | X\langle ac\rangle) P(X\langle ac\rangle | X\langle abc\rangle) P(X\langle abc\rangle). \end{aligned} \tag{6}$$

The conditional distributions in equation (6) can be derived straightforwardly from the full joint distribution; for example:$P(X\langle a\rangle|X\langle ac\rangle) = \frac{P(X\langle a\rangle, X\langle ac\rangle)}{P(X\langle ac\rangle)}$ where the distributions on the right hand side are marginals of the full joint distribution.

We refer to a factored distribution like that in equation (6) determined by a general rollup ρ as P_ρ. Thus summarizing the above discussion in formal terms for the general case, we have:

Definition 5 *Given a set of breakdowns B, and a joint distribution P on $X\langle\mathcal{S}(B)\rangle$, a roll-up $\rho \in \Psi_B$ determines a distribution P_ρ on the set of random variables $X\langle V_\rho\rangle$ given by:*

$$P_\rho(X\langle V_\rho\rangle) = \prod_{b \in V_\rho} P(X\langle b\rangle|X\langle \mathrm{pa}_\rho(b)\rangle), \tag{7}$$

where for $c \neq 0$, the conditional distributions are derived from P:

$$P(X\langle b\rangle|X\langle c\rangle) = \frac{P(X\langle b\rangle, X\langle c\rangle)}{P(X\langle c\rangle)}, \tag{8}$$

and by convention, $P(X\langle b\rangle|X\langle 0\rangle) = P(X\langle b\rangle)$.

4.1 Characterizing the Optimal Roll Up

The previous section has shown how every roll up on a set of breakdowns B induces a probability distribution P_ρ on a subset of breakdowns in the semilattice $\mathcal{S}(B)$ generated by B. In general, P_ρ is an approximation to the actual joint probability distribution of the random variables in the roll up, and again in general, the some roll-ups will provide better approximations than others. If it is to be a reliable guide to the actual data, a roll-up should provide a good approximation to its actual distribution, and so it is not unreasonable to characterize the “optimal” roll-up as that with the best approximation amongst all alternatives. Thus we have:

Postulate 1 *The* optimal *roll-up on a set of breakdowns B is that roll-up $\rho \in \Psi_B$ whose associated factored distribution, P_ρ, most closely approximates (in a sense to be defined) the actual joint probability distribution, $P(X\langle V_\rho\rangle)$, of its associated random variables.*

For a formal quantification of the degree of approximation in the above, we again follow Chow and Liu (1968) in using the *Kullback-Leibler distance*, $D_{\mathrm{KL}}\{P \,\|\, Q\}$, which for probability mass functions P and Q defined on a common sample space Ξ is defined: $D_{\mathrm{KL}}\{P \,\|\, Q\} = \sum_{\xi \in \Xi} P(\xi) \log \frac{P(\xi)}{Q(\xi)}$. Informally, $D_{\mathrm{KL}}\{P \,\|\, Q\}$ measures the information contained in the probability distribution P that is not determined by Q. As e.g. Cover and Thomas (2006) demonstrate, it is always greater than or equal to 0, with equality iff P and Q are the same distribution. Thus we can restate postulate (1) formally as follows:

Definition 6 *Given a set of breakdowns B, and a joint distribution P on $X\langle\mathcal{S}(B)\rangle$, an* optimal *roll-up is given as:*

$$\operatorname*{argmin}_{\rho \in \Psi_B} D_{\mathrm{KL}}\{P_\rho \,\|\, P(X\langle V_\rho\rangle)\}. \tag{9}$$

Use of the Kullback-Leibler distance in this application yields a particularly appealing characterization of optimal roll-ups by way of a generalization of Chow and Liu’s (1968) work. To state the result, recall that the *entropy*, $H(X)$, of a discrete random variable X is defined: $H(X) = -\sum_X P(X) \log P(X)$. Similarly, the *mutual information*, $I(X;Y)$, between discrete random variables X and Y is defined: $I(X;Y) = \sum_X \sum_Y P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)}$. We adopt the standard conventions that $0 \log 0 = 0$, $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$, so that both $H(X)$ and $I(X;Y)$ are always positive. Now it can be shown (see appendix A for the proof) that:

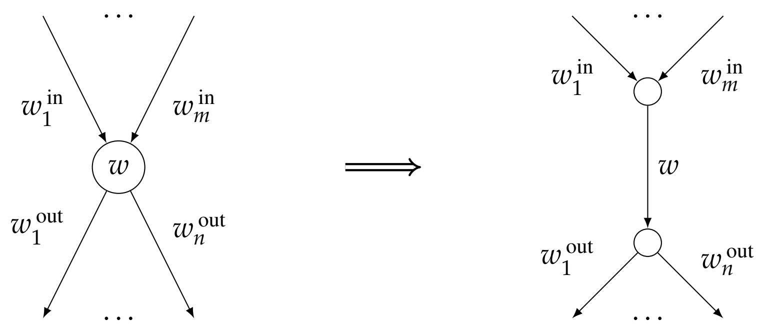

Figure 5: Elimination of node weights.

Theorem 1 *For a set B of breakdowns, an equivalent characterization of an optimum roll-up according to definition 6 is:*

$$\underset{\rho \in \Psi_B}{\operatorname{argmin}} \sum_{b \in V_\rho} H(X\langle b\rangle) - I(X\langle b\rangle; X\langle \mathrm{pa}_\rho(b)\rangle). \tag{10}$$

Regarding the above, for some roll-up $\rho \in \Psi_B$, consider the terms $H(X\langle b\rangle)$ and $-I(X\langle b\rangle; X\langle \mathrm{pa}_\rho(b)\rangle)$ for some $b \in V_\rho$. Viewing ρ as a Steiner arborescence in accordance with its definition in (3), $H(X\langle b\rangle)$ may be considered as a weight associated with the node $X\langle b\rangle$, and $-I(X\langle b\rangle; X\langle \mathrm{pa}_\rho(b)\rangle)$ as another (negative) weight associated with the arc from $\mathrm{pa}_\rho(b)$ to b in ρ. The summation in equation (10) is therefore the total node and arc weight of the arborescence ρ, and the theorem states that to find the optimum roll-up, we must find the arborescence whose total weight is minimal amongst all roll-ups on $X\langle \mathcal{S}(B)\rangle$. In the vernacular of the literature, we seek a *minimum weight node-weighted Steiner arborescence*.

5. Implementation

The first task in implementing the approach described in the previous sections is that of generating the dual order graph, $\mathcal{G}(B)$, on a given set of breakdowns, B. For this purpose, we have found the algorithm described by Chase (1971) effective. Unfortunately, the problem we face in obtaining an optimal roll-up—namely, that of computing a minimum-weight Steiner arborescence on the resulting graph—is known to be (strongly) NP-hard, as e.g. Hwang and Richards (1992) observe. Authors including Charikar et al. (1999) and Zelikovsky (1997) have proposed *polynomial time approximation schemes* (PTAS) for the problem, which are guaranteed to produce arborescences whose total weight is within a specified ratio of the optimum (indeed the simple expedient of connecting each terminal to the root by way of the shortest path is one such scheme, though by no means the best performing).

In our particular application, however, we have found an adaptation of a heuristic technique due to Stanojević and Vujošević (2006) suffices to solve the problem *exactly* in all instances we have encountered to date. Stanojević and Vujošević's (2006) approach is an operational simplification of the branch-and-cut technique Koch and Martin (1998), and tackles the problem of minimal Steiner trees on *un*directed graphs; here, we adapt the approach to Steiner arborescences on directed graphs.

5.1 Problem reduction

We proceed by noting first that a simple manipulation of the dual order graph eliminates node weights from the problem. As illustrated in figure 5, each weighted node in the dual order graph (a typical such node, with weight w is depicted on the left of the figure) is replaced (as shown on the right) by by two unweighted nodes, connected by a single arc

$$
\begin{aligned}
\text{minimize} \qquad & \sum_{a \in A} w_a y_a && && (11)\\
\text{subject to} \qquad & \sum_{a \in \delta^-(M)} y_a \geq 1 && \forall M \subseteq V.\ M \cap B \neq \emptyset \text{ and } r \in V \backslash M, && (12)\\
& y_a \in \{0, 1\} && \forall a \in A.
\end{aligned}
$$

Figure 6: An binary integer programming formulation of the Steiner arborescence problem.

with the same weight. It is straightforward to show that any Steiner arborescence on the original graph corresponds to a unique Steiner arborescence on the new graph with the same weight, and so from a minimum weight arborescence on the new graph, we can construct a minimum weight arborescence on the original.

The edge weights of the directed graph resulting from the above step comprise both the positive and negative terms in the summation of equation (10). Negative edge weights can be inconvenient to deal with using the integer programming approach described in the next section (they necessitate the addition of further constraints to ensure that solutions are indeed trees), and so we eliminate them in turn by subtracting the (generally negative) minimum edge weight from the weight of each edge.

5.2 Binary integer program formulation

Given a set of breakdowns B, the upshot of applying above reduction steps to the dual order graph generated by B is a directed graph (V, A) with a set of nodes V including a root node r, a set of arcs A and for each $a \in A$, a positive weight w_a. The breakdowns in $B \subseteq V$ constitute the set of terminal vertices in $\mathcal{G}$. Recall that the optimum roll-up connects all elements of B to the root node r with minimal total edge weight.

Figure 6 shows how the problem may be formulated as a binary integer program (BIP). The Steiner tree we seek is characterized by a collection of binary variables, $\{y_a | a \in A\}$; y_a is 1 iff the arc a is to be included in the tree. The objective function in equation (11) sums the weights of all edges in the tree, as would be expected. The collection of constraints in (12) uses *graph cuts* to stipulate that there exists a path in the solution from the root to each node in the terminal set. Each *cut* M comprises a set of nodes that includes at least one terminal node, but excludes the root. A cut determines a *cutset*, $\delta^-(M) = \{a \in A \mid \mathrm{tl}(a) \in V_\rho \backslash M$ and $\mathrm{hd}(a) \in \delta^-(M)\}$, the set of all arcs from nodes outside of M to nodes in M. Here, for any arc a from node v to v', we use the expressions $\mathrm{tl}(a)$ and $\mathrm{hd}(a)$ to denote v and v', respectively. The constraints in (12) thus require that a least one edge in the solution crosses every graph cut separating the root from the terminals.

Unfortunately, the presentation in figure 6 (which—strictly speaking—is a program *schema*) is deceptively brief, since (12) represents an exponential number of constraints. Not every such constraint, however, will be binding or *active* in a solution to the program, and only those constraints which are active need actually be imposed. Since it is impossible to predict *a priori* which constraints are indeed active in a solution, we begin with a relaxation of the full problem involving a small subset of the graph cut constraints. Upon solving the relaxed problem, we examine the solution to see if solves the full problem; this involves verifying that the root is connected to all the terminals as required by the full set of cutset contraints. If indeed the solution does solve the full problem, we are done: Imposing further constraints is bootless, and cannot further reduce the weight of the arborescence. Otherwise, further cut constraints are added to the relaxed problem, and the process is repeated.

Algorithm 1 An algorithm for Steiner arborescences based on binary integer programming.

— *Given a (n expanded) dual order graph* $\mathcal{G}(B)$ *generated by a set of breakdowns* B*, weighted in accordance with equation* (10) *and the reduction of section 5.1, return a minimum-weight Steiner tree on* $\mathcal{G}(B)$ *with terminal nodes given by* B*.*

1: Formulate an initial binary integer program, with the objective function given in (11), but no constraints.
2: ROOTS $\leftarrow B$.
3: **while** ROOTS $\neq \{r\}$ **do** ▹ *Solution does not yet represent a Steiner arborescence.*
4:     **for** each $b \in$ ROOTS$\backslash\{r\}$ **do**
5:         Add constraints $\sum_{a\in\delta^-(D)} y_a \geq 1$ to the program, where D is b and the set of all nodes connected to b by edges in the current solution.
6:     **end for**
7:     Solve the resulting BIP, yielding $Y^* = \{y_a^* \mid a \in A\}$.
8:     ROOTS $\leftarrow \{\mathrm{tl}(y_a^*) \mid y_a^* = 1\}\backslash\{\mathrm{hd}(y_a^*) \mid y_a^* = 1\}$.
9: **end while**
10: **return** Y^*

Pseudo-code for the implementation is given in algorithm 1. Driving the addition of cutset constraints is the set ROOTS. This is initially the given set of breakdowns—the terminal nodes of the desired Steiner arborescence. In lines 4 to 6, cutset constraints are added for each member of ROOTS. Solution in line 8 of the resulting (relaxed) binary integer program is discussed below. With a solution to the relaxed program in hand, the set ROOTS is set to the "roots" of the solution—those nodes in the solution that are descended from no other solution nodes. For the solution to be a Steiner arborescence, the roots of the solution should coincide with the (sole) root, r, of the dual order graph; if not, new cutset constraints are added forcing the next solution closer to r.

Of course, solving the binary integer program in line 8 is itself far from trivial in general. Fortunately, as Meindl and Templ (2012) observe, a number of effective software packages for mixed integer programming are freely available on the Internet. While the performance of such no-cost packages still lags that of commercial offerings, like Stanojević and Vujošević, we have found the open source package lpSolve (Berkelaar et al. 2013) quite capable of solving the BIPs that arise in practical applications of algorithm 1.

6. Conclusions

In the foregoing, we have set out a framework for organizing random variables associated with breakdowns in a data cube. Taking our lead from Chow and Liu (1968), we showed how *roll-ups*—gathering variables in increasing levels of aggregation—can be considered as graphical specifications of joint probability distributions, which hold with greater or less degrees of accuracy in the data. Finding the most accurate such approximation is a Steiner arborescence problem, which we have found can be tackled effectively in this instance using integer programming techniques.

We have used the problem of organizing anomaly reports throughout the paper both as a motivating application and an illustrative example of the framework proposed. It should be reasonably straightforward, however, to apply the framework a number of other contexts in which a collection of random variables is associated with breakdowns in a data cube.

Examples of such situations come easily to mind: Sales over time, for instance, across product lines and sales regions, disease reports in geographic regions within demographic groups, species counts within classified biomes and so on.

The development in this paper assumes that all variables are discrete, allowing for very direct estimation of the entropy and mutual information values involved in theorem (1). In principle, it should be possible to extend the treatment to continuous random variables using differential formulations of entropy and mutual information, as Suzuki (2012) demonstrates; estimation of these quantities becomes significantly more challenging, however, and it becomes necessary to take steps to avoid over-fitting (Suzuki proposes a version of the Chow-Liu approach based on the minimum description length criterion, using Bayesian estimators).

In section 3.3, we described a procedure for computing the values of unobserved random variables associated with aggregate breakdowns from the values of random variables associated with their downsets. Again, there is great latitude to substitute alternate procedures within the framework to accomodate random variables with other interpretations. It is vital, however, that the values of these synthesized random variables be determined non-stochastically by the values of their constituents, since a critical step in the proof of theorem (1) rests on the assumption that their conditional entropy given their constituents is 0 (see appendix A).

A. Proof of Theorem 1

Abbreviate $X_\rho = X\langle V_\rho \rangle, \mathrm{pa}(b) = \mathrm{pa}_\rho(b)$. Ten:

$$
\begin{aligned}
D_{\mathrm{KL}}\{P_\rho \,\|\, P(X_\rho)\} &= \sum_{X_\rho} P(X_\rho) \log \frac{P(X_\rho)}{P_\rho(X_\rho)} \\
&= -\sum_{X_\rho} P(X_\rho) \log P_\rho(X_\rho) + \sum_{X_\rho} P(X_\rho) \log P(X_\rho) \\
&= -\sum_{X_\rho} P(X_\rho) \sum_{b\in V} \log P_\rho(X\langle b\rangle | X\langle \mathrm{pa}(b)\rangle) + \sum_{X_\rho} P(X_\rho) \log P(X_\rho) \\
&= -\sum_{X_\rho} P(X_\rho) \sum_{b\in V} \log \frac{P(X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle)}{P(X\langle \mathrm{pa}(b)\rangle)} + \sum_{X_\rho} P(X_\rho) \log P(X_\rho)
\end{aligned}
$$

and adding and subtracting $\sum_{X_\rho} P(X_\rho) \sum_{b\in V} \log P(X\langle b\rangle)$,

$$
\begin{aligned}
&= -\sum_{X_\rho} P(X_\rho) \sum_{b\in V} \log \frac{P(X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle)}{P(X\langle b\rangle) P(X\langle \mathrm{pa}(b)\rangle)} - \sum_{X_\rho} P(X_\rho) \sum_{b\in V} \log P(X\langle b\rangle) \\
&\qquad + \sum_{X_\rho} P(X_\rho) \log P(X_\rho) \\
&= \sum_{b\in V} \Big[-\sum_{X_\rho} P(X_\rho) \log \frac{P(X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle)}{P(X\langle b\rangle) P(X\langle \mathrm{pa}(b)\rangle)} - \sum_{X_\rho} P(X_\rho) \log P(X\langle b\rangle) \Big] \\
&\qquad + \sum_{X_\rho} P(X_\rho) \log P(X_\rho).
\end{aligned}
$$

Now observe that for $b \in V$:

$$
\begin{aligned}
-\sum_{X_\rho} P(X_\rho)\log P(X\langle b\rangle) &= -\sum_{X\langle b\rangle}\sum_{X\langle V\setminus\{b\}\rangle} P(X\langle b\rangle)P(X\langle V\setminus\{b\}\rangle|X\langle b\rangle)\log P(X\langle b\rangle)\\
&= -\sum_{X\langle b\rangle}\Big[P(X\langle b\rangle)\log P(X\langle b\rangle)\sum_{X\langle V\setminus\{b\}\rangle} P(X\langle V\setminus\{b\}\rangle|X\langle b\rangle)\Big]\\
&= -\sum_{X\langle b\rangle} P(X\langle b\rangle)\log P(X\langle b\rangle)\\
&= H(X\langle b\rangle).
\end{aligned}
$$

Similarly:

$$
\sum_{X_\rho} P(X_\rho)\log\sum_{b\in V}\frac{P(X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle)}{P(X\langle b\rangle)P(X\langle \mathrm{pa}(b)\rangle)} = \sum_{X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle} P(X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle)\log\frac{P(X\langle b\rangle, X\langle \mathrm{pa}(b)\rangle)}{P(X\langle b\rangle)P(X\langle \mathrm{pa}(b)\rangle)} \tag{13}
$$

$$
= I(X\langle b\rangle; X\langle \mathrm{pa}(b)\rangle). \tag{14}
$$

And by definition, $\sum_{X_\rho} P(X_\rho)\log P(X_\rho) = -H(X_\rho)$. Rewriting equation (13), therefore:

$$
D_{\mathrm{KL}}\{P_\rho \,\|\, P(X_\rho)\} = \sum_{b\in V}\Big[-I(X\langle b\rangle; X\langle \mathrm{pa}(b)\rangle) + H(X\langle b\rangle)\Big] - H(X_\rho). \tag{15}
$$

With regard to the term $H(X_\rho)$, note that for a rollup ρ on breakdowns B, since the variables $X\langle V_\rho \backslash B\rangle$ are (non-stochastically) determined by the values of $X\langle B\rangle$, by the chain rule of entropy (Cover and Thomas 2006, p. 22), $H(X\langle V_\rho\rangle) = H(X\langle B\rangle)$. Thus from equation (15), to minimize $D_{\mathrm{KL}}\{P_\rho \,\|\, P(X_\rho)\}$ for a given set of breakdowns B, we must choose a rollup such that $\sum_{b\in V} H(X\langle b\rangle) - I(X\langle b\rangle; X\langle \mathrm{pa}(b)\rangle)$ is minimal amongst corresponding quantities for all rollups on B; formally:

$$
\operatorname*{argmin}_{\rho\in\Psi_B} D_{\mathrm{KL}}\{P_\rho \,\|\, P(X\langle V_\rho\rangle)\} = \operatorname*{argmin}_{\rho\in\Psi_B}\sum_{b\in V_\rho} H(X\langle b\rangle) - I(X\langle b\rangle; X\langle \mathrm{pa}_\rho(b)\rangle). \tag{16}
$$

References

D. Agarwal, D. Barman, D. Gunopulos, N. Young, F. Korn, and D. Srivastava. Efficient and effective explanation of change in hierarchical summaries. In *Proc. of the 13th ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining*, KDD '07, pages 6–15, 2007.

R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 232–243, 1997.

M. Berkelaar, K. Eikland, and P. Notebaert. lp_solve reference guide, 2013. URL `http://lpsolve.sourceforge.net/`.

M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed steiner problems. *Jnl. of Algorithms*, 33(1):73–91, 1999.

P. J. Chase. Efficient subsemilattice generation. *Jnl. Comb. Th., Series A*, 10(2):181–182, 1971.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Th.*, 14(3):462–467, 1968.

C. K. Chow and T. Wagner. Consistency of an estimate of tree-dependent probability distribution. *IEEE Trans. Inf. Th.*, 19(3):369–371, 1973.

E. F. Codd. A relational model of data for large shared data banks. *Comm. ACM*, 13(6): 377–387, June 1970.

E. F. Codd, S. B. Codd, and C. T. Salley. Providing OLAP (On-line Analytical Processing) to user analysts: An IT mandate. Technical report, Codd & Date, Inc., 1993.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.

R. Fagin, R. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Multi-structural databases. In *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Sys.*, PODS '05, pages 184–195, 2005a.

R. Fagin, Ph. Kolaitis, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Efficient implementation of large-scale multi-structural databases. In *Proc. of the 31st Intl. Conf. on Very Large Data Bases*, VLDB '05, pages 958–969, 2005b.

M. X. Goemans and Y.-S. Myung. A catalog of Steiner tree formulations. *Networks*, 23(1): 19–28, 1993.

J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, and M. Venkatro. Data Cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–53, 1997.

M. Gyssens and L. V. S. Lakshmanan. A foundation for multi-dimensional databases. In *Proc. of the 23rd Intl. Conf. on Very Large Data Bases*, VLDB '97, pages 106–115, 1997.

V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proc. 1996 ACM SIGMOD Intl. Conf. on Management of Data*, SIGMOD '96, pages 205–216, 1996.

F. K. Hwang and D. S. Richards. Steiner tree problems. *Networks*, 22(1):55–89, 1992.

T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.

T. Koski and J. Noble. *Bayesian Networks: An Introduction*. Wiley, 1st edition, 2009.

P. McCullagh. Exchangeability and regression models. In A. C. Davison, Y. Dodge, and N. Wermuth, editors, *Celebrating Statistics: Papers in Honour of Sir David Cox on his 80th Birthday*, chapter 4, pages 89–110. Oxford University Press, 2005.

B. Meindl and M. Templ. Analysis of commercial and free and open source solvers for linear optimization problems. Technical report, Technische Universität Wien, 2012.

S. Sarawagi. Explaining differences in multidimensional aggregates. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 42–53, 1999.

M. Stanojević and M. Vujošević. An exact algorithm for Steiner tree problem on graphs. *Intl. Jnl. of Computers, Comm. & Control*, 1(1):41–46, 2006.

J. Suzuki. The Bayesian Chow-Liu algorithm. In *Proc. 6th European Workshop on Probabilistic Graphical Models*, 2012.

R. Torlone. Conceptual multidimensional models. In M. Rafanelli, editor, *Multidimensional Databases: Problems and Solutions*, chapter 3, pages 69–90. Idea Group Publishing, 2003.

A. Zelikovsky. A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, 18(1):99–110, 1997.